fgm inc

Garlic Fries
Track Broker and Archiver

DII COE Developers' Technical Exchange

May 24, 2000

Ellen Minderman
FGM Inc.
ellen.minderman@fgm.com

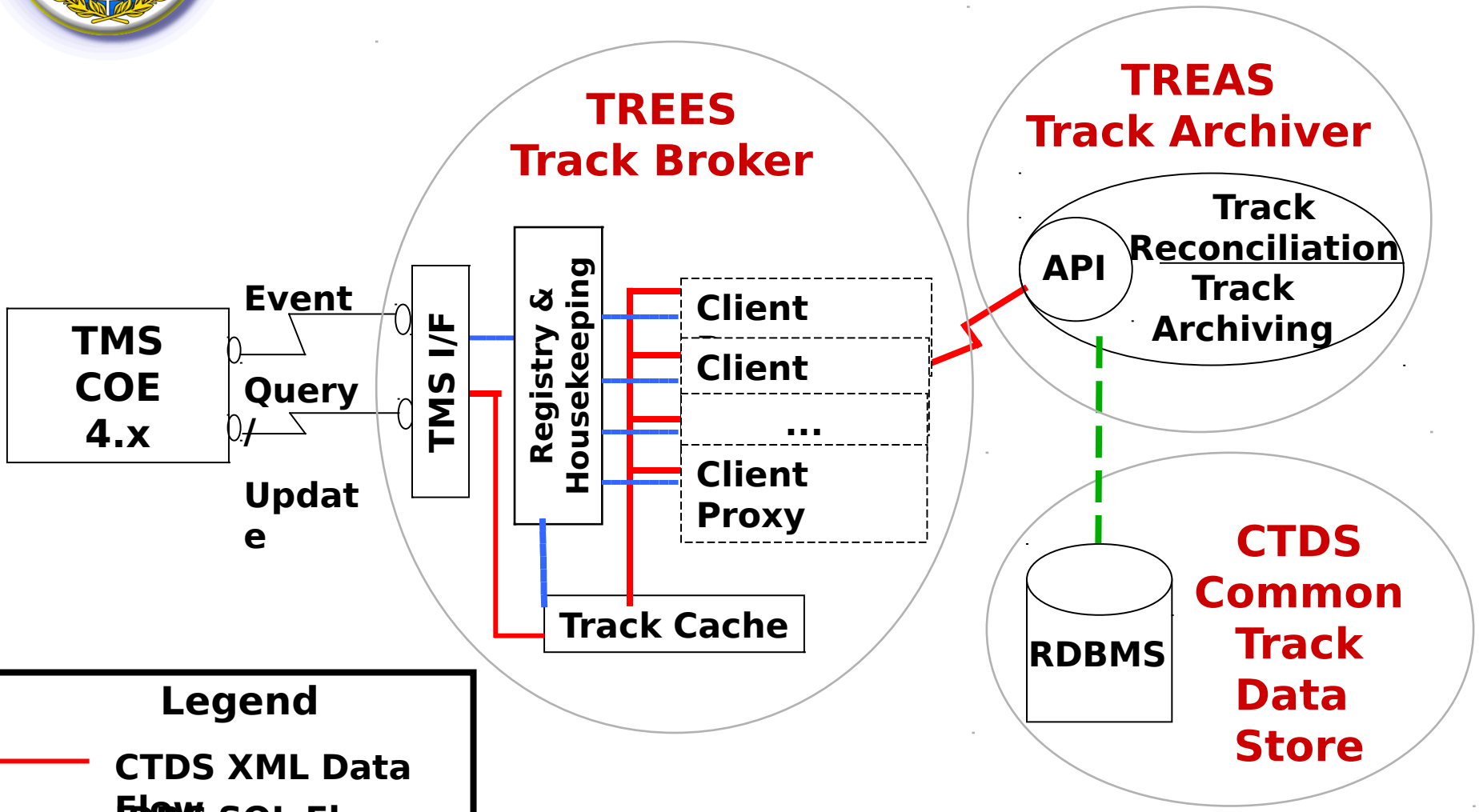
Contents

- Goals
- GF Data Engineering Components Overview
- Track Broker
- Track Archiver
- Common Track Data Store

Goals

- **2-way data exchange between ICSF track data management and mission-specific DBMSs**
  - Enables archiving of historical data
  - Eases integration with additional information stored in DBMS
  - Improved ability to link mission apps to tracks
- **Design Goals**
  - Decouple TMS from Archivers
  - Employ platform-neutral XML data encoding
  - Achieve good portability (via Java, RMI & C/JNI where required)

Garlic Fries Architecture
Data Engineering Components



Track Broker

TREES - Track RElational Extension Server

- Provides enhanced track event distribution services
  - Transaction API – request/response protocol
  - Subscription API – on-going event by event protocol
- Native TMS format re-packaged as self-describing XML-based tracks
  - Mitigates archiver exposure to evolving track data formats
  - XML Parser provides simplified extraction of desired track data by archivers
- Accessible by archivers over WAN & LAN, via provided client support library
  - Synchronous, archiver maintains persistent connection
  - Asynchronous, archiver performs aperiodic connection

Track Archiver

TREAS - Track RElational Archiver

- Reference implementation
  - Java source code provided in SDK
- Historical store of track data using RDBMS
- Standards-based, vendor independent interfaces (JDBC)
- Track reconciliation
- Stores track and amplifying data in the Common Track Data Store (CTDS)
  - Amplifying data may be associated with tracks and reports
  - Superset of TMS and non-TMS CTDS data elements

Common Track Data Store (CTDS)

- **DB Segment for Track Identity and Location Reports**
  - Schema definition was a collaboration between I3, TMS, TMV and Shade
  - Uses surrogate keys for primary keys, stored procedures provided for surrogate key generation.
  - Triggers used to manage relationships between tables
- **Also stores amplifying information**
  - Acoustic and Elint Characteristics of the contact provided by the sensor
  - Information about the mission that identified the contact
  - Reaction of the observed contact
  - Any observed damage to the contact

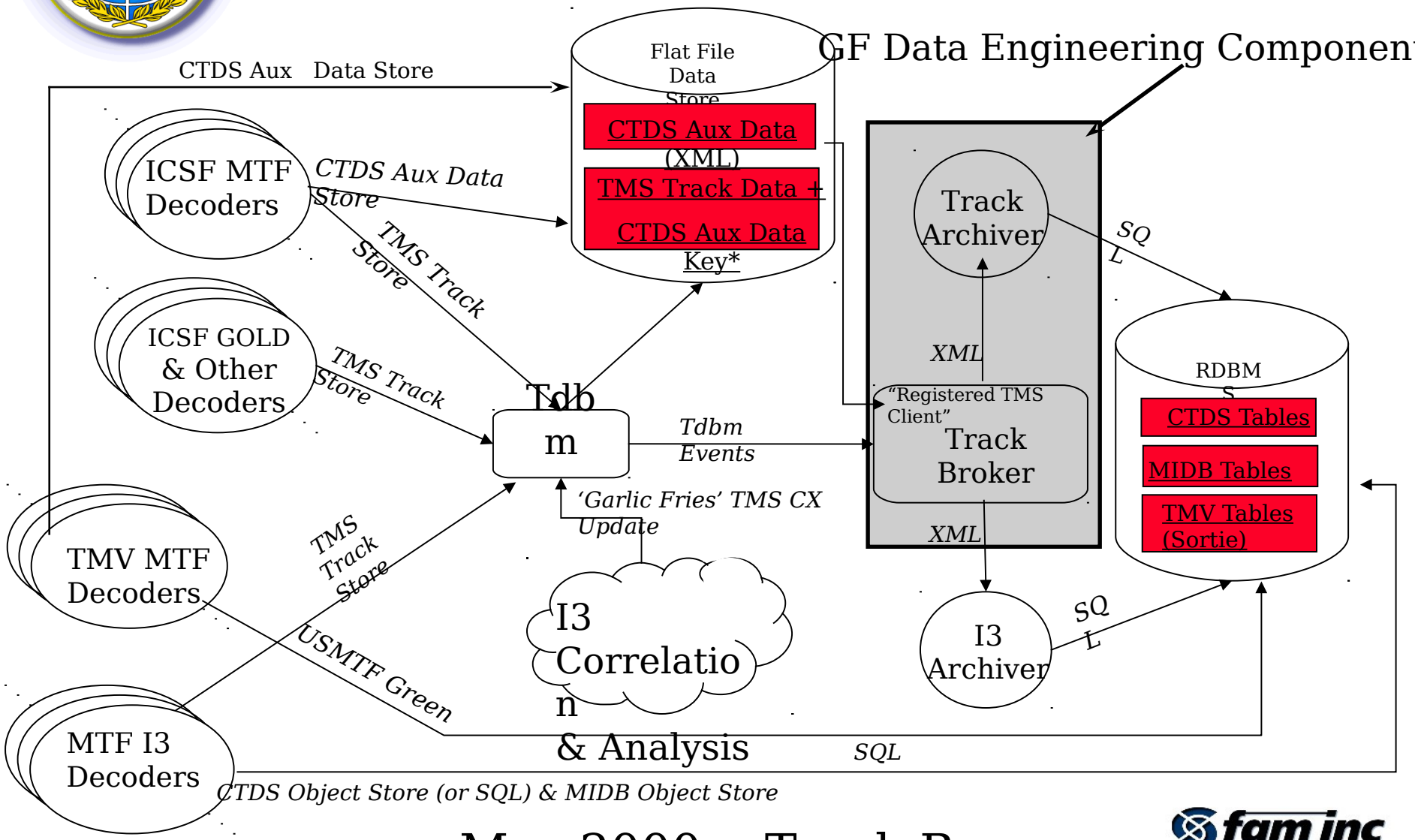
Developer Tools

- **Archiver Software Development Kit (SDK)**
  - Archiver Source code
  - TREES API libraries & documentation
    - Java Docs for Java Broker API
    - XML DTD Specification
    - Programmers Manual
  - Support libraries & documentation
    - Includes IBM XML Java Parser
- **CTDS documentation**
  - Mapping of CTDS database schema to messages
  - ERWIN model
  - DBDD documentation

Broker XML Document Format

- **Provides interface to track and amplifying data**
  - Broker Subscription Interface – event by event
  - Broker Transaction Interface – request/response
- **Based on CTDS Database Schema**
  - Tables are XML elements
  - Columns also XML elements
- **XML DTD Provided**
  - Available from the XML Registry
- **Caveats**
  - No data validation
  - Auxiliary Documents must be error free
  - Error Handling

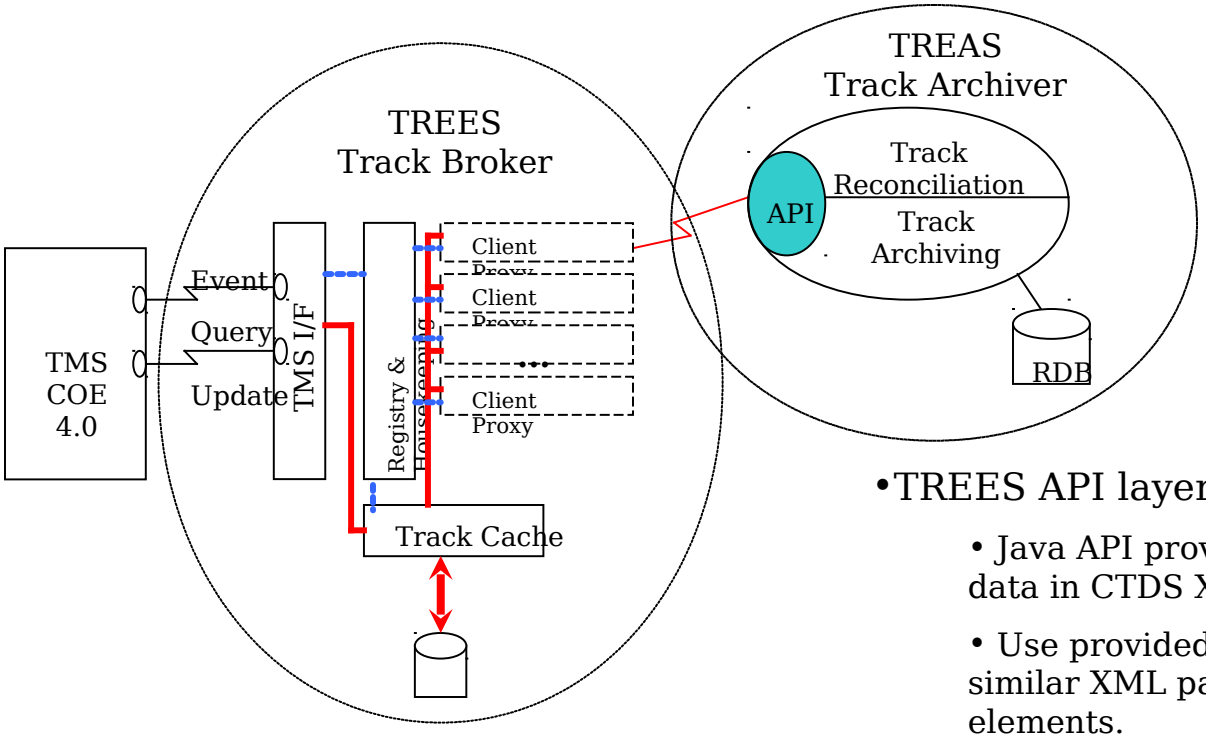
Track Broker/Archiver as a GCCS-M Message Processing Component



Questions?

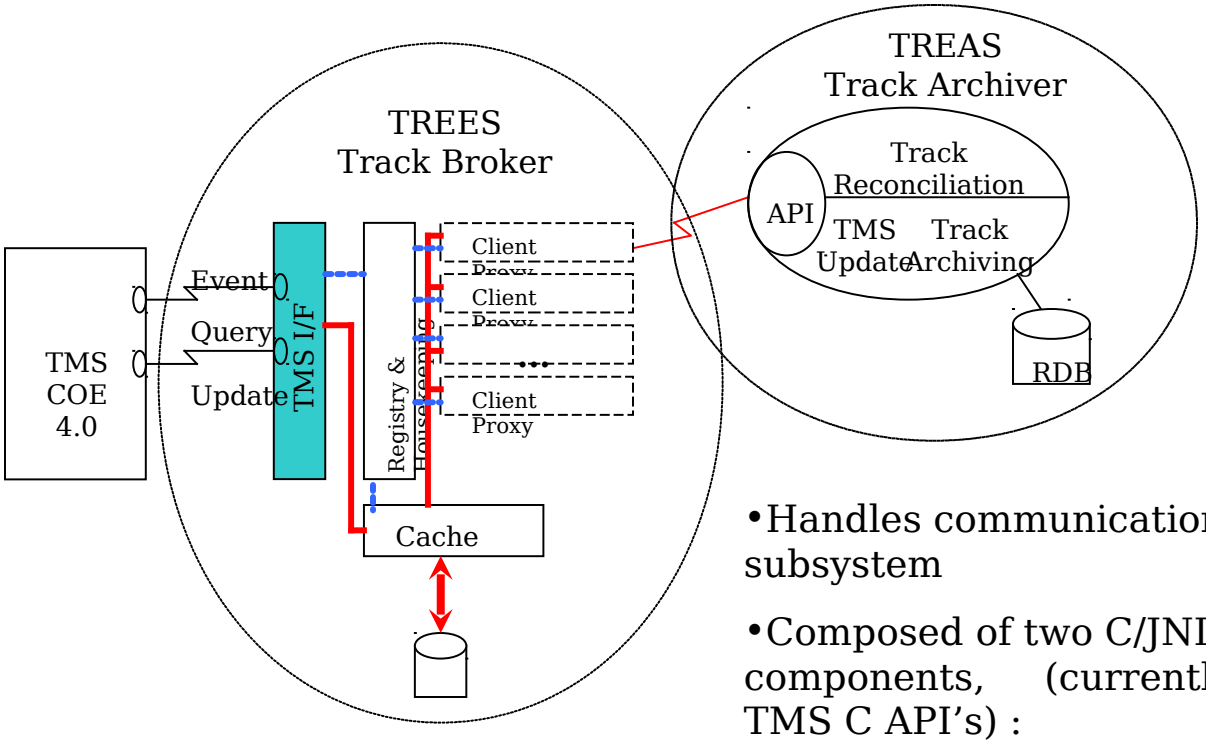
Backup

Garlic Fries Architecture
TREES API



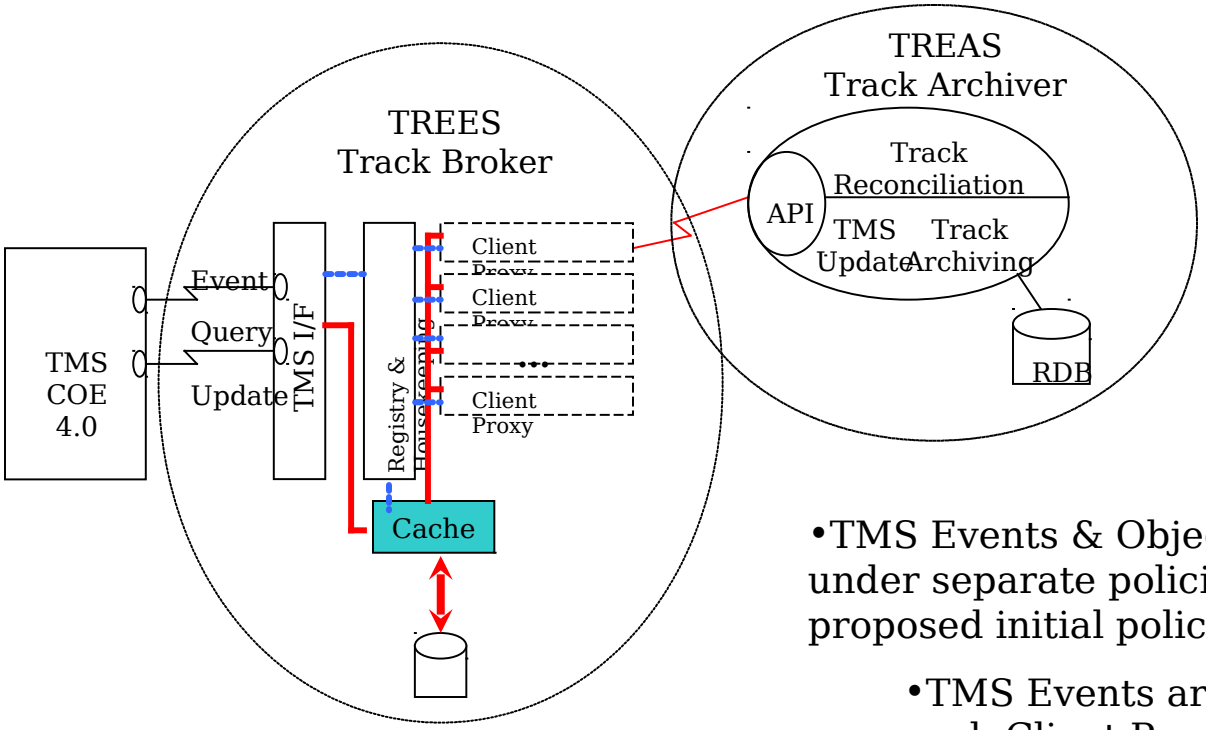
- TREES API layer -
  - Java API provides access to TMS data in CTDS XML Format.
  - Use provided IBM XML4J or similar XML parser to extract data elements.
- API provides both event by event and transaction interfaces, and will offer future update interface.
- TREAS Archiver source provides example use of TREES API.

TREES
TMS I/F



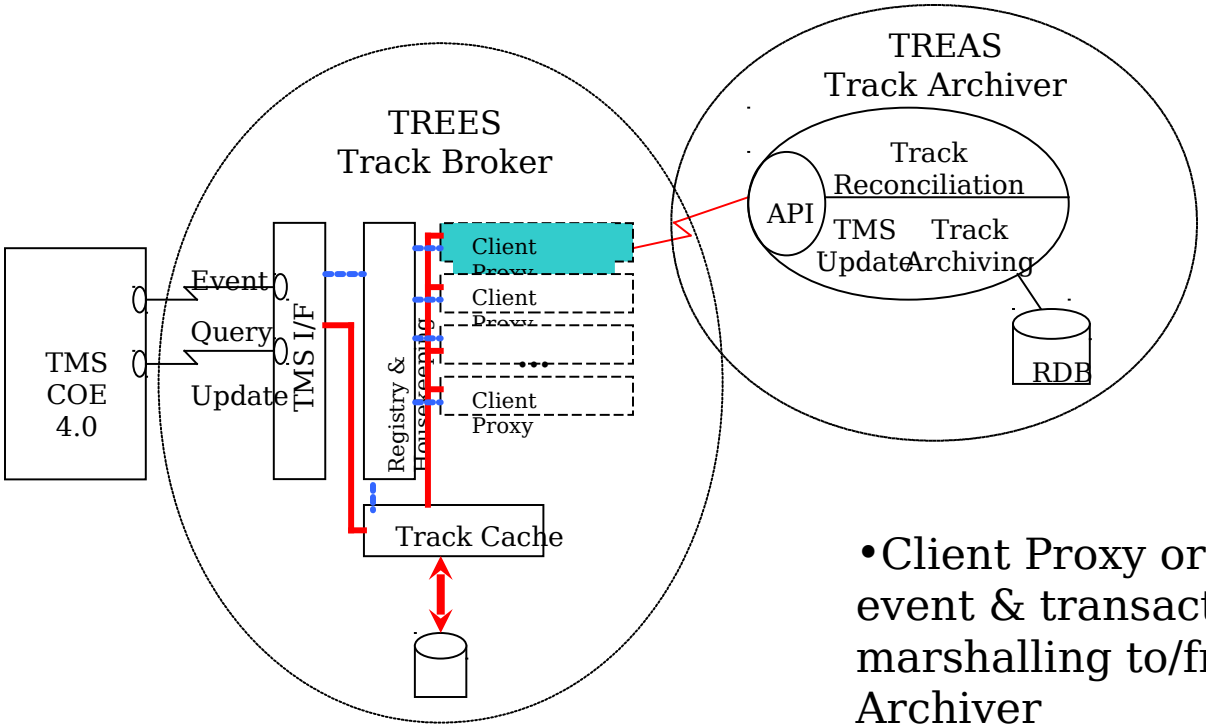
- Handles communications with TMS subsystem
- Composed of two C/JNI/Java based components, (currently interfacing to TMS C API's) :
  - pTms handles event polling
  - tTms handles transactions (query & update)
- C code generates XML docs for TMS events & objects (such as Tracks) which are then stored in the Cache via Java/RMI

TREES
Cache



- TMS Events & Objects are cached under separate policies; our proposed initial policies:
  - TMS Events are offered to each Client Proxy, then deleted from cache
  - TMS Tracks are offered to each Client Proxy, then cached for re-vending. Tracks are updated in cache by new events
  - Cache is always cleared on

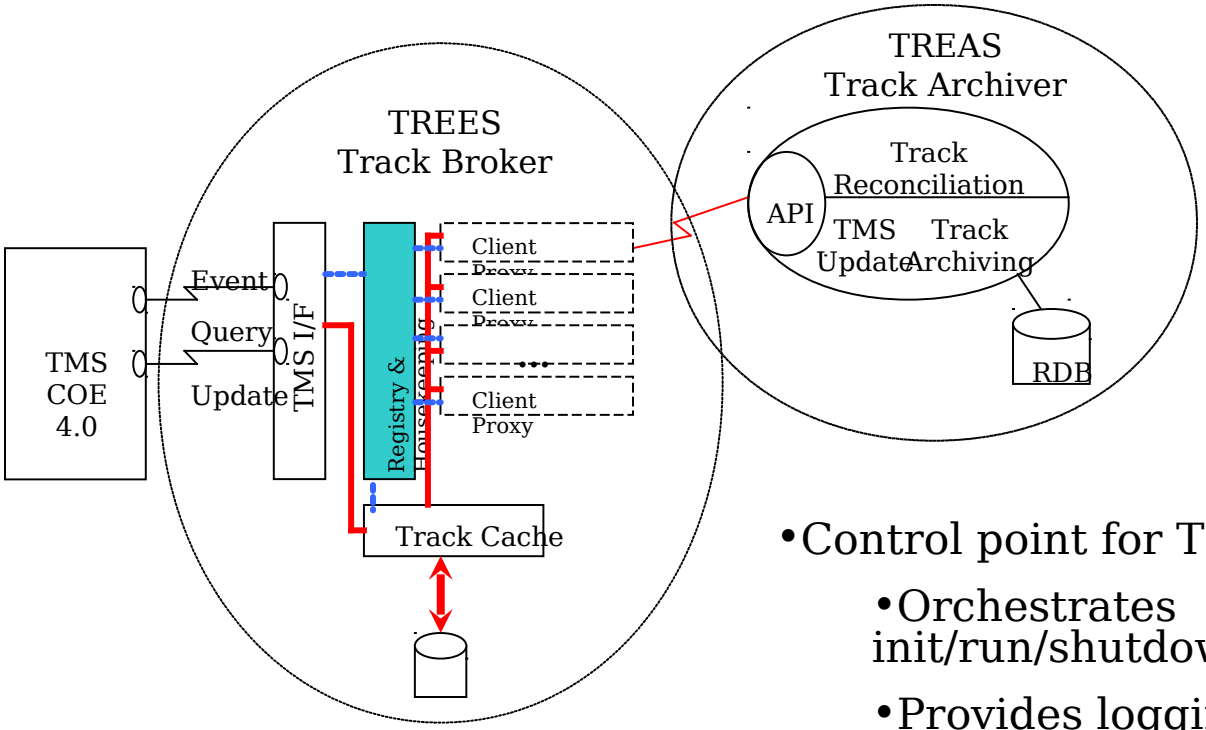
TREES
Client Proxy



•Client Proxy orchestrates event & transaction marshalling to/from TREAS Archiver

fgm inc

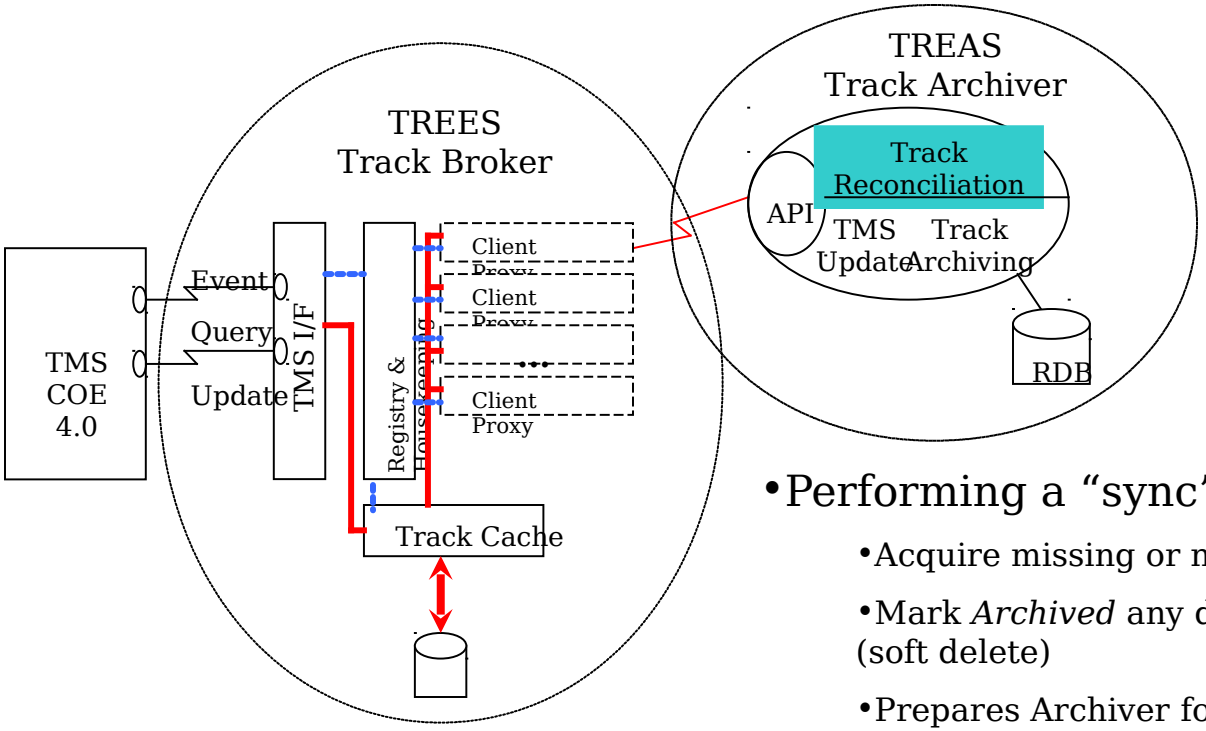
TREES
Registry & Housekeeping



- Control point for TREES Broker
  - Orchestrates init/run/shutdown cycle
  - Provides logging/configuration services
  - Provides central RMI registry
- Embedded FESI (ECMA Script) Interpreter available for configuration/policy mods.

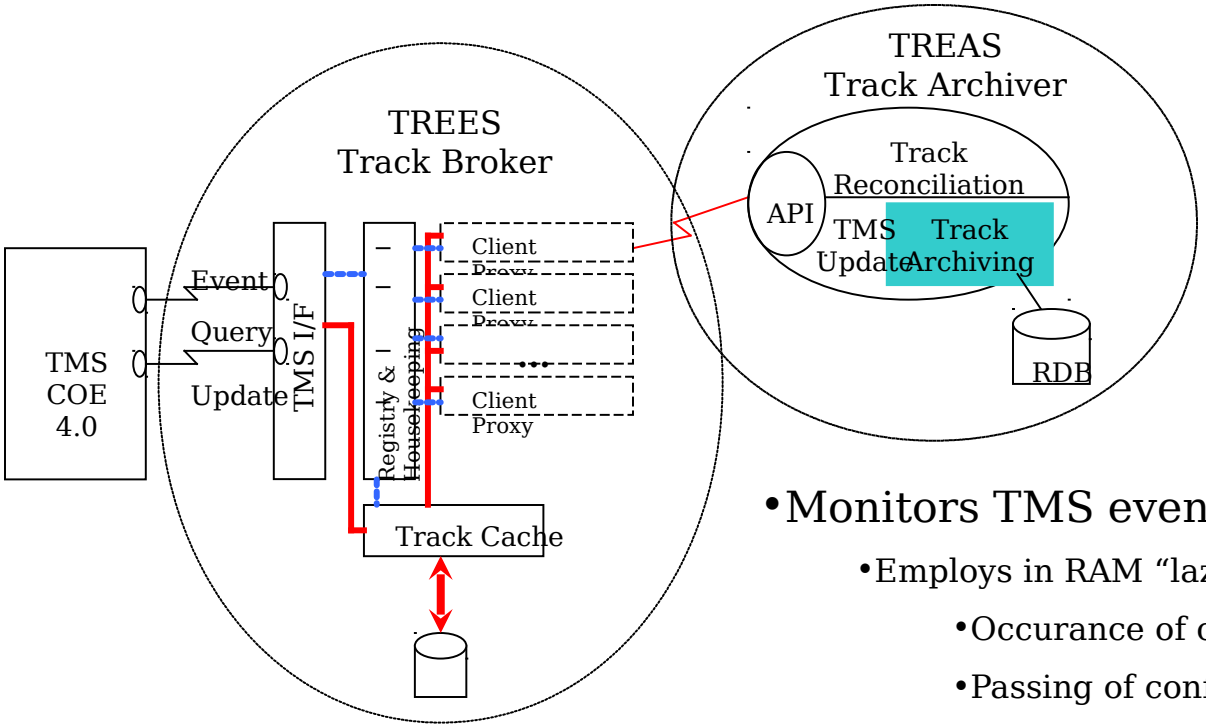
May 2000 -- Track Br

TREAS
Track Reconciliation



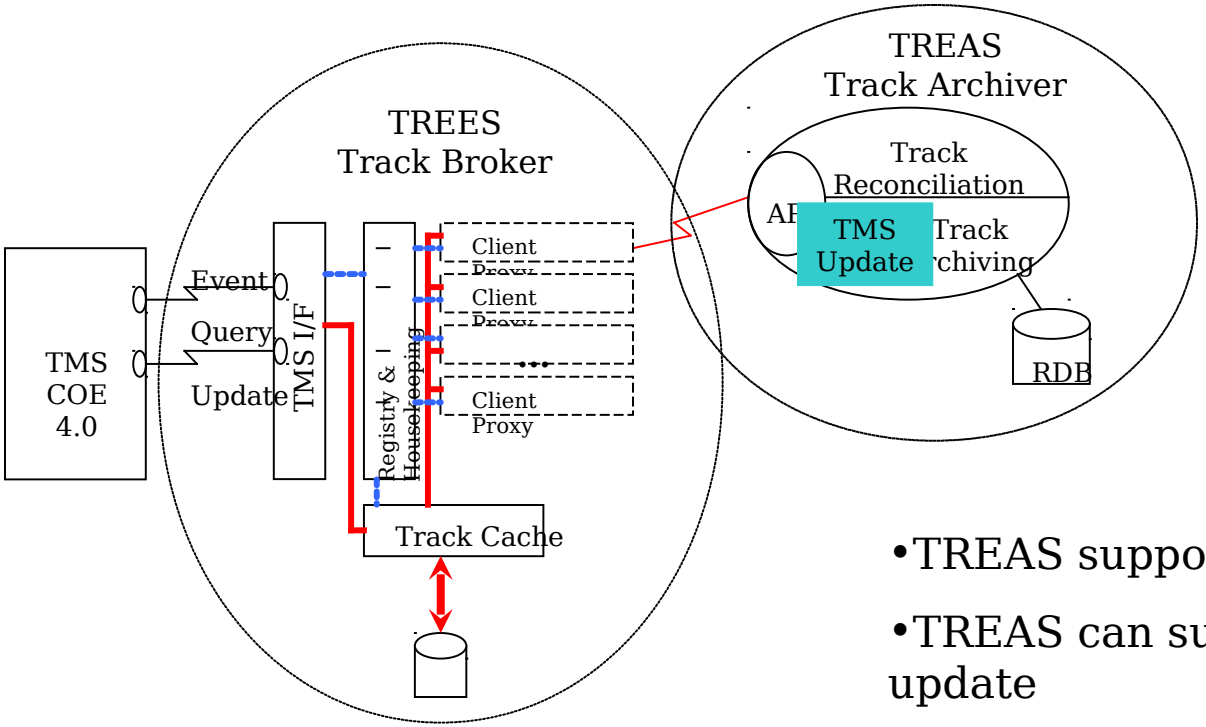
- Performing a "sync" with TMS
  - Acquire missing or new data
  - Mark *Archived* any data TMS has dropped (soft delete)
  - Prepares Archiver for handling new incoming events
- Uses TREES Java Object API to talk to Broker
- Uses JDBC to talk to RDBMS

TREAS
Track Archiving



- Monitors TMS event stream from Broker
  - Employs in RAM "lazy write" strategy, with flush
    - Occurance of configurable number of event
    - Passing of configurable time interval
- Uses TREES Java Object API to talk to Br
- Uses JDBC to talk to RDBMS

TREAS
TMS Update



- TREAS supports UIC lookup
- TREAS can support UIC update